ECサイト研修システム

新卒研修向けのテスト技法学習用ECサイトアプリケーションです。

🎯 目的

このシステムは以下のテスト技法を実践的に学ぶために作成されています:

- APIのテスト
- SQL文を使ったDBの中身の確認
- 同値分割
- 境界値分析
- デシジョンテーブル
- 状態遷移図・状態遷移表
- 組合せテスト(直交法、オールペア法)

🛠 技術スタック

- **フロントエンド**: React 18.2
- **バックエンドAPI**: Google Apps Script
- **データベース**: Google Spreadsheet
- **ホスティング**: GitHub Pages
- **UIライブラリ**: lucide-react

📁 プロジェクト構成

```
ec-training-site/
├── public/
│   └── index.html          # HTMLベース
├── src/
│   ├── pages/
│   │   ├── ProductListPage.js  # 商品一覧ページ
│   │   ├── ProductListPage.css
│   │   ├── CartPage.js         # カートページ
│   │   ├── CartPage.css
│   │   ├── CheckoutPage.js     # チェックアウトページ
│   │   ├── CheckoutPage.css
│   │   ├── OrderHistoryPage.js # 注文履歴ページ
│   │   └── OrderHistoryPage.css
│   ├── services/
```

```
│   │   └── api.js          # API通信サービス
│   ├── App.js              # メインアプリケーション
│   ├── App.css
│   ├── index.js            # エントリーポイント
│   └── index.css           # グローバルスタイル
├── package.json
└── README.md
```

🚀 セットアップ手順

1. Google Spreadsheet & GASのセットアップ

1. Google Driveで新しいスプレッドシートを作成
2. 以下のシートを作成:
   - products (商品)
   - members (会員)
   - cart_items (カート)
   - orders (注文)
   - order_items (注文明細)
   - coupons (クーポン)
3. 「拡張機能」→「Apps Script」を開く
4. 提供されたGASコードを貼り付け
5. `SPREADSHEET_ID` を実際のIDに変更
6. `initializeData` 関数を実行して初期データを投入
7. 「デプロイ」→「ウェブアプリとしてデプロイ」
8. アクセス権限を「全員」に設定
9. デプロイURLをコピー

2. Reactアプリのセットアップ

```bash
```

```bash
# リポジトリをクローン
git clone https://github.com/YOUR_USERNAME/ec-training-site.git
cd ec-training-site

# 依存関係をインストール
npm install

# src/services/api.js のGAS_API_URLを更新
# 'YOUR_GAS_DEPLOYMENT_URL_HERE' を実際のURLに置き換え

# ローカルで起動
npm start
```

3. GitHub Pagesにデプロイ

```bash
# package.json の homepage を更新
# "homepage": "https://YOUR_USERNAME.github.io/ec-training-site"

# gh-pagesをインストール
npm install --save-dev gh-pages

# デプロイ
npm run deploy
```

🎨 主要機能

商品一覧

- カテゴリフィルター
- 商品検索
- 在庫状況表示
- カート追加

ショッピングカート

- 数量変更
- 商品削除
- 送料計算(5000円以上で無料)
- 小計・合計表示

チェックアウト

- 支払方法選択(クレジット/代引き/コンビニ/銀行振込)
- 配送方法選択(通常/速達/日時指定)
- クーポン適用
- 注文確認

注文履歴

- 過去の注文一覧
- 注文詳細表示
- ステータス確認

🧪 テストシナリオ例

同値分割のテスト

```
購入数量:
- 有効クラス: 1個, 5個, 在庫数
- 無効クラス: 0個, -1個, 在庫数+1
```

境界値分析のテスト

```
送料無料ライン(5000円):
- 4999円 → 送料500円
- 5000円 → 送料0円
- 5001円 → 送料0円
```

APIテストの例

```bash
# 商品一覧取得
curl "YOUR_GAS_URL?path=products&method=GET"

# カートに追加
curl -X POST "YOUR_GAS_URL?path=cart&method=POST" \
  -d '{"member_id":1,"product_id":1,"quantity":2}'
```

📊 データベース構造

products (商品)

- id, name, price, stock, category, description, age_restriction, status

members (会員)

- id, email, password_hash, name, birth_date, rank, status, points, created_at

cart_items (カート)

- id, member_id, product_id, quantity, added_at

orders (注文)

- id, member_id, status, total_amount, discount_amount, shipping_fee, payment_method, shipping_method, shipping_address, created_at, updated_at

order_items (注文明細)

- id, order_id, product_id, quantity, unit_price, subtotal

coupons (クーポン)

- id, code, discount_type, discount_value, min_purchase, valid_from, valid_until, usage_limit, used_count

🐛 トラブルシューティング

モックモードで動作する

`src/services/api.js` の `GAS_API_URL` が `'YOUR_GAS_DEPLOYMENT_URL_HERE'` のままの場合、自動的にモックデータで動作します。実際のGAS URLに置き換えてください。

CORSエラー

GASのデプロイ設定で「アクセスできるユーザー」が「全員」になっているか確認してください。

商品が表示されない

1. GASで `initializeData` 関数を実行したか確認
2. ブラウザの開発者ツールでネットワークタブを確認
3. APIレスポンスを確認

📝 ライセンス

このプロジェクトは研修目的で作成されています。

🤝 サポート

質問や問題がある場合は、Issueを作成してください。